

Fig.2

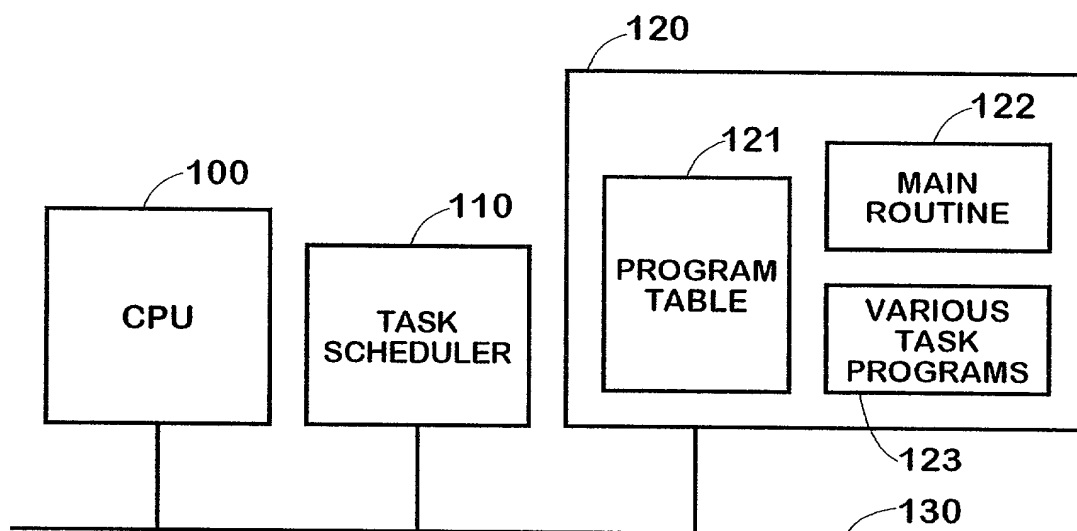


Fig.3

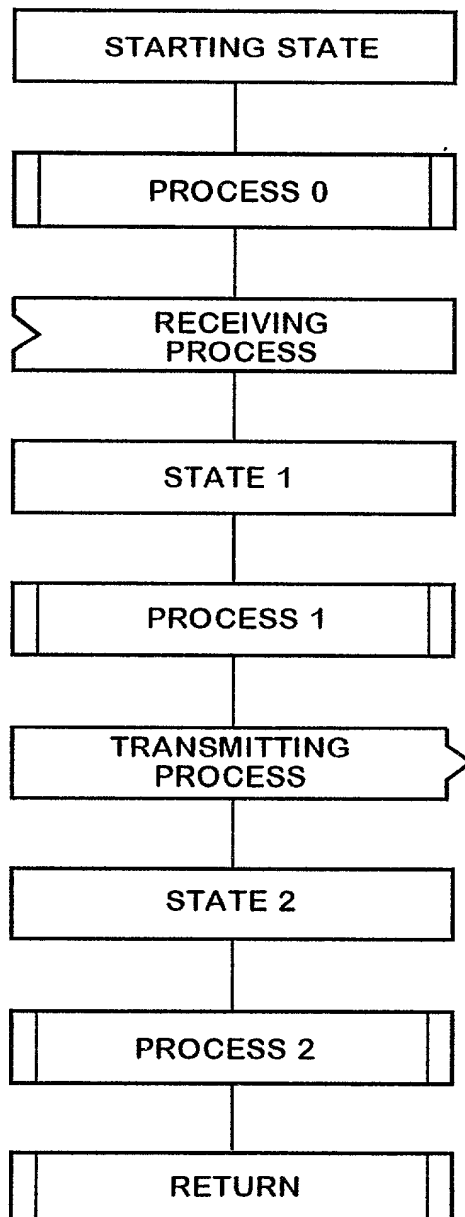


Fig.4

```
func()
{
    proc0:
        Process Contents 0;
    recieve(chanel0, data);
    proc1:
        Process Contents 1;
    send(chanel0, data);
    proc2:
        Process Contents 2;
    return;
}
```

Fig.5A

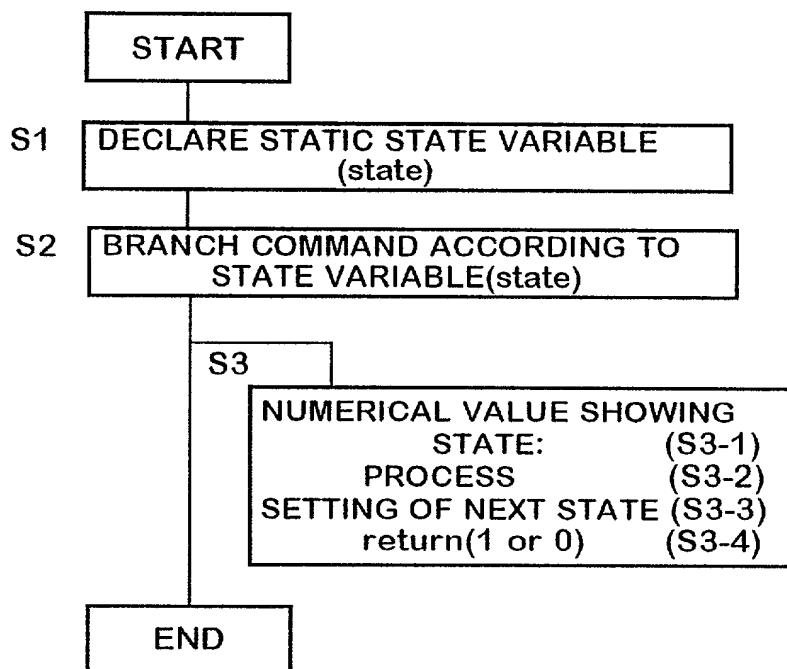


Fig.5B

```
static int state; // S1
func()
{
    switch(state&0x3) { // S2
        //S3
        case0: //S3-1
            Process Contents 0; // S3-2
            state=1; // S3-3
            return(0); // S3-4
        case1:
            get(chanel0,data);
            Process Contents 1;
            send(chanel0, data);
            state=2;
            return(0);
        case2:
            Process Contents 2;
            state=0;
            return(0);
        defaults:
            state=0;
            return(0);
    }
}
```

Fig.6A

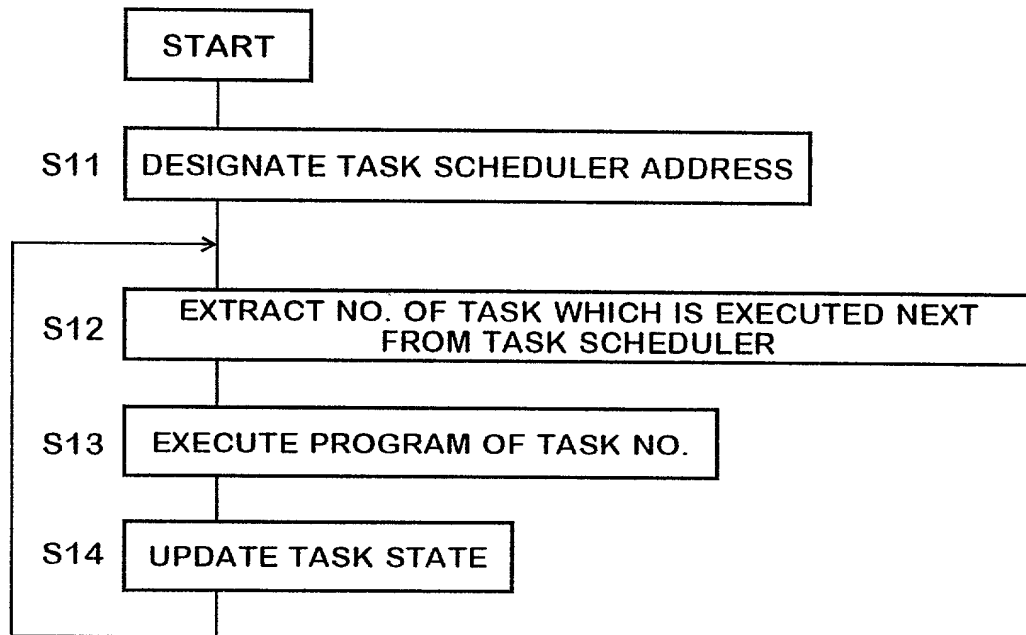


Fig.6B

```
int. *sp;  
sp= TASK_SCHEDULER_ADR; // S11  
  
while(1) {  
    NUM= *sp; // S12  
    state[NUM]=func[NUM](); // S13&S14  
}
```

0940850 05804660

Fig.7A

task0 b110_101
task1 b011_110
task2 b011_011

ADDRESS OF EACH TASK

Fig.7B

	7	6	5	4	3	2	1	0
7	0	0	0	0	0	0	0	0
6	0	0	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
3	0	1	0	0	1	0	0	0
2	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

SITUATION OF 8 × 8 BIT MATRIX

Fig.8A

```

static int sta0;

int task0()
{
    switch(sta0&0x1) { //13_S
    case0:
        sta0=1;
        return(0);
    case1:
        sta0=0;
        return(0);
    }
}

```

CONTENS OF TASK 0

Fig.8B

```

static int sta1;

int task1()
{
    switch(sta1&0x1) { //19_S
    case0:
        sta1=1;
        return(1);
    case1:
        sta1=0;
        return(0);
    }
}

```

CONTENS OF TASK 1

Fig.8C

```

static int sta2;

int task2()
{
    switch(sta2&0x1) { //26_S
    case0:
        state(task0_id)=1;
        sta2=1;
        return(1);
    case1:
        sta2=0;
        return(0);
    }
}

```

CONTENS OF TASK 2

Fig.9

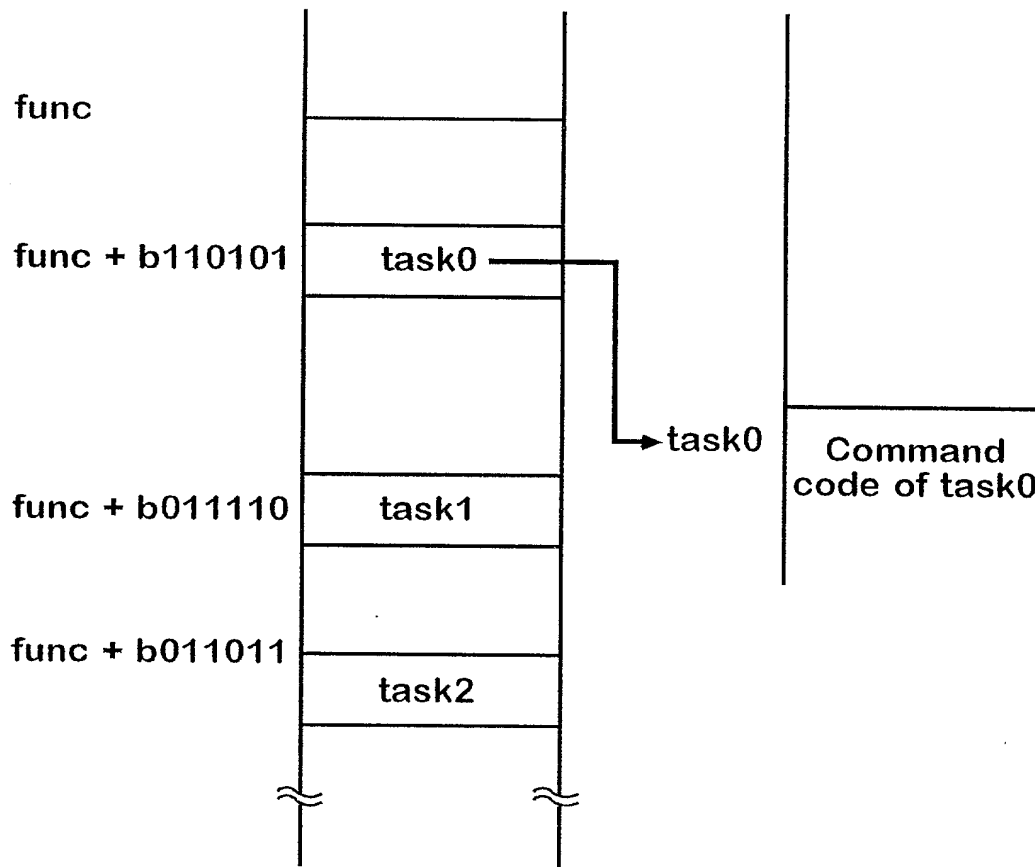


Fig.10

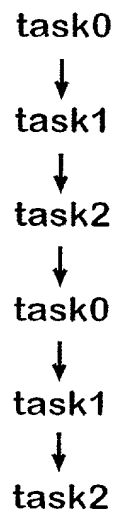


Fig.11

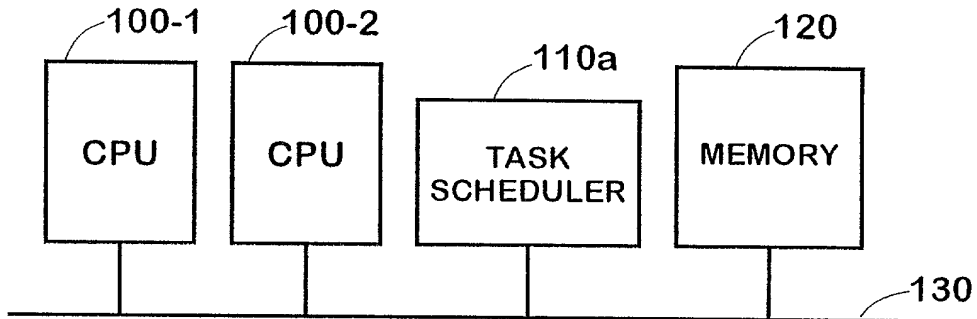


Fig.12

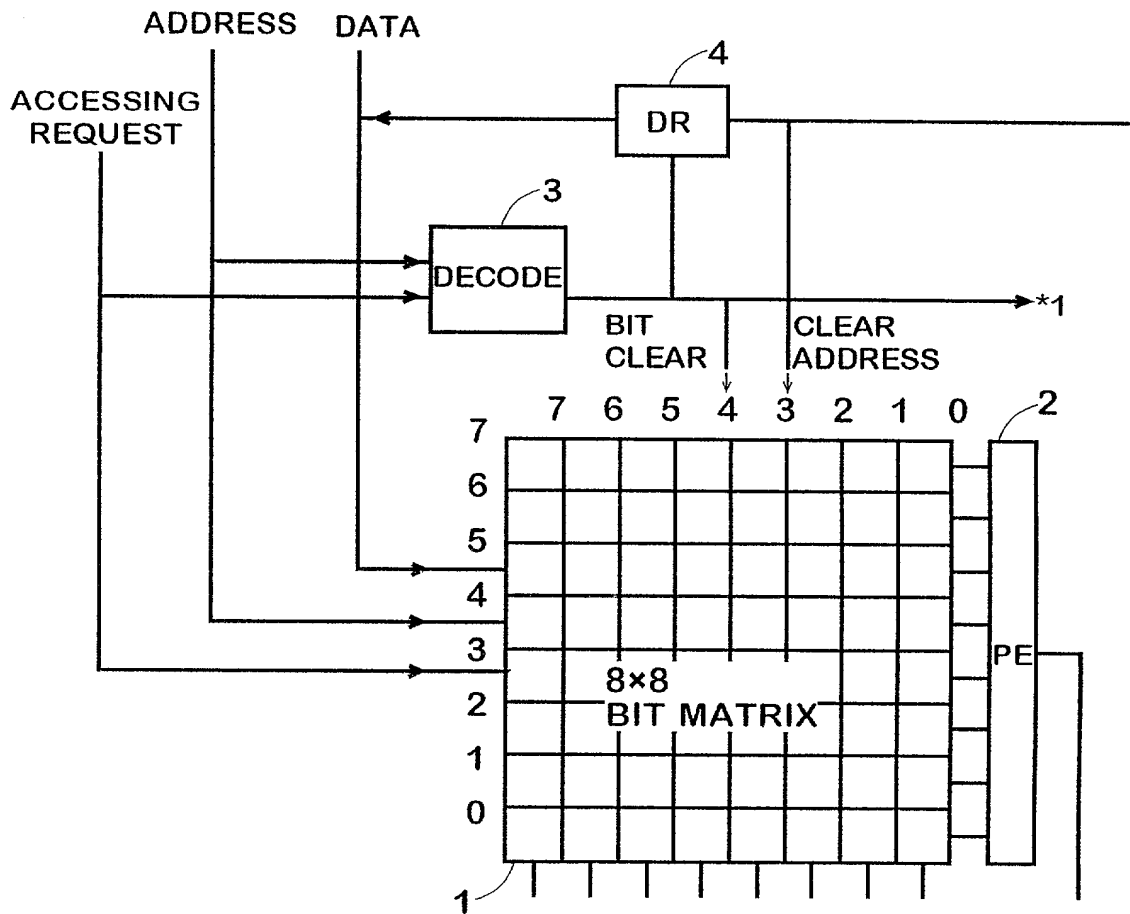


Fig.13

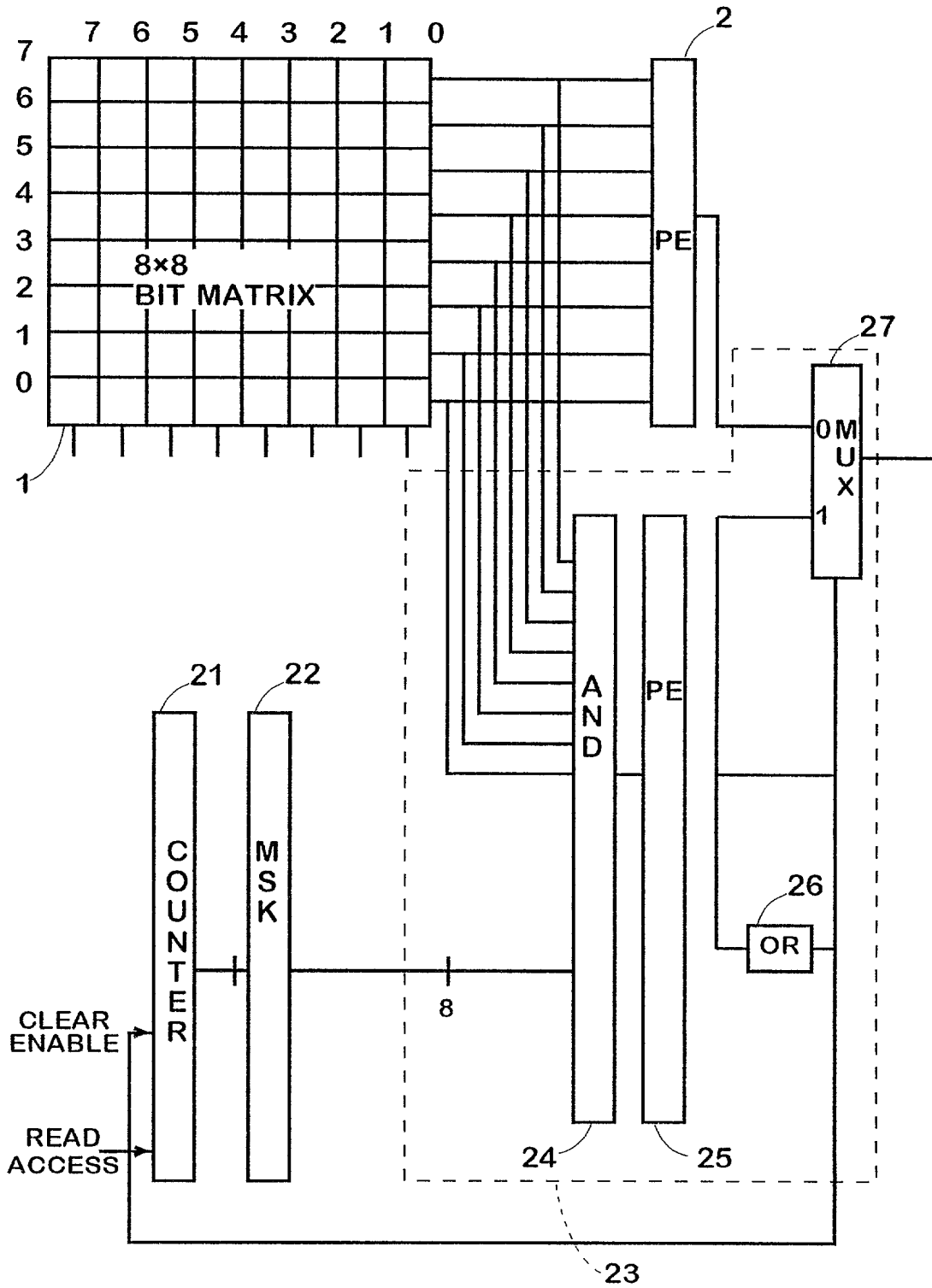


Fig. 14

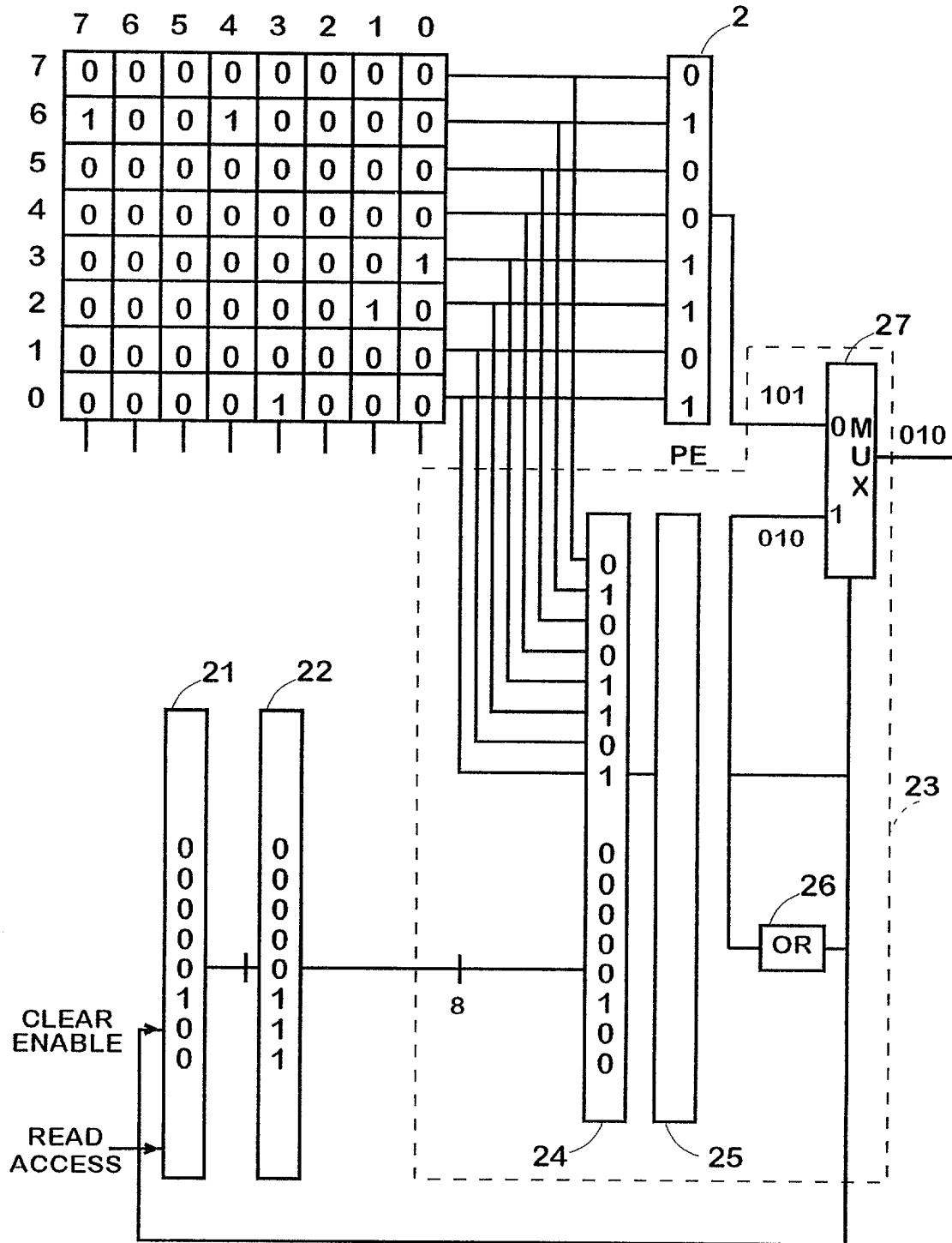


Fig. 15

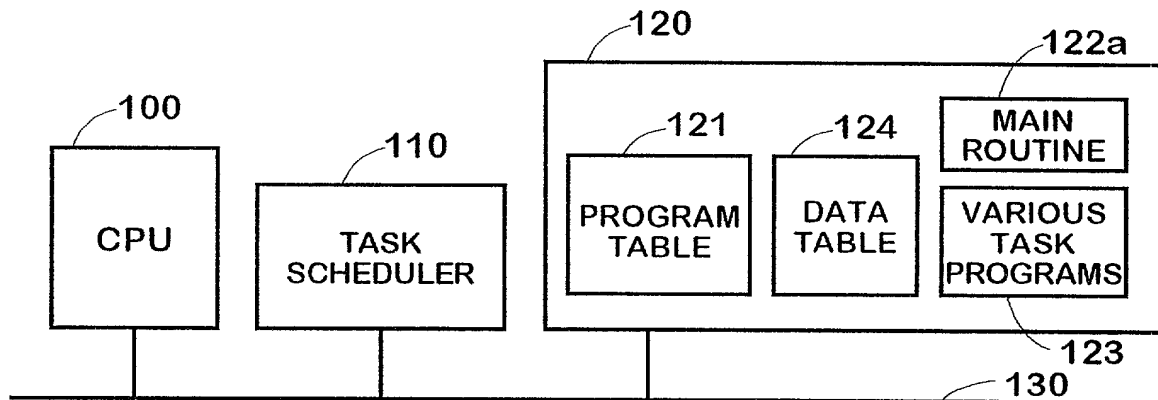


Fig. 16

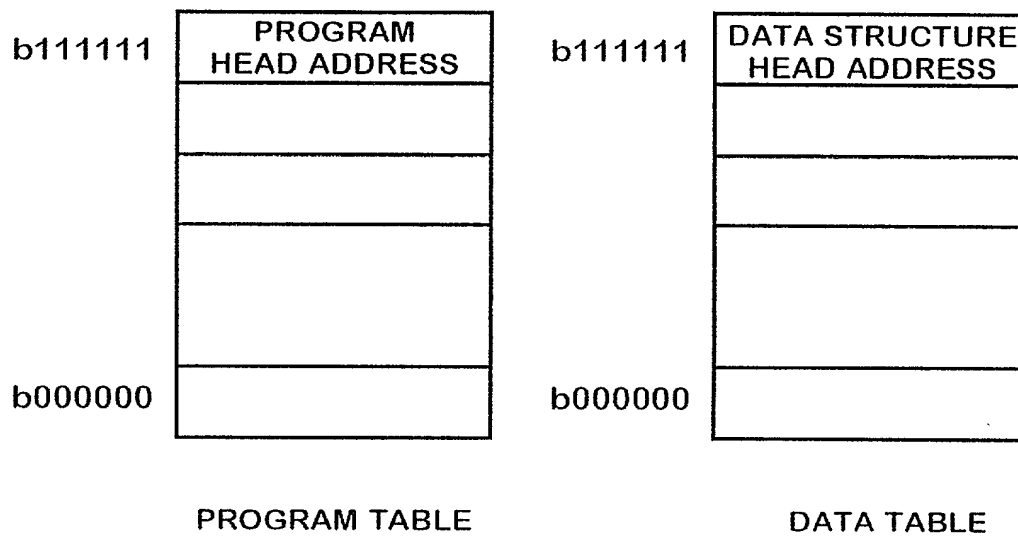


Fig.17A

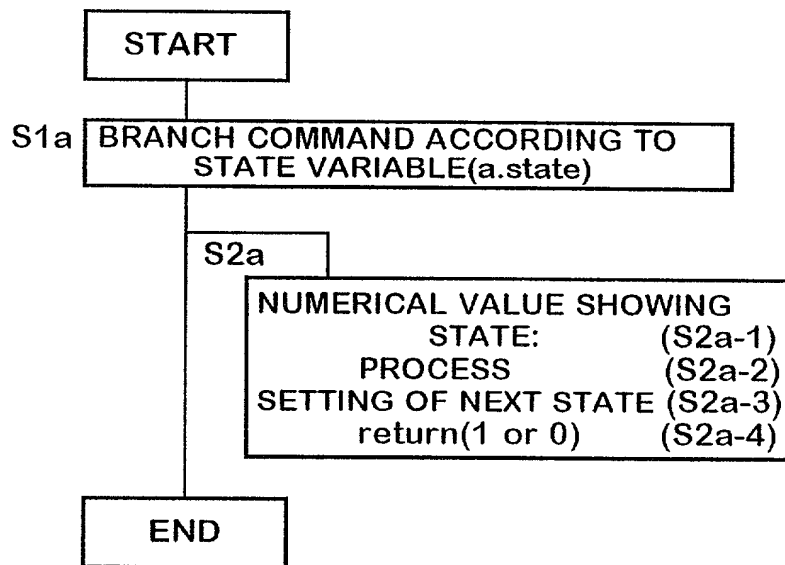


Fig.17B

```
int. func(struct xxx*a)
{
    switch(a.state&0x3) { // S1a
    //S2a
    case0: //S2a-1
        Process contents 0; // S2a-2
        state=1; // S2a-3
        return(0); // S2a-4
    case1:
        get(chanel0,data);
        Process contents 1;
        send(chanel0, data);
        a.state=2;
        return(0);
    case2:
        Process contents 2;
        a.state=0;
        return(0);
    defaults:
        a.state=0;
        return(0);
    }
}
```

Fig.18A

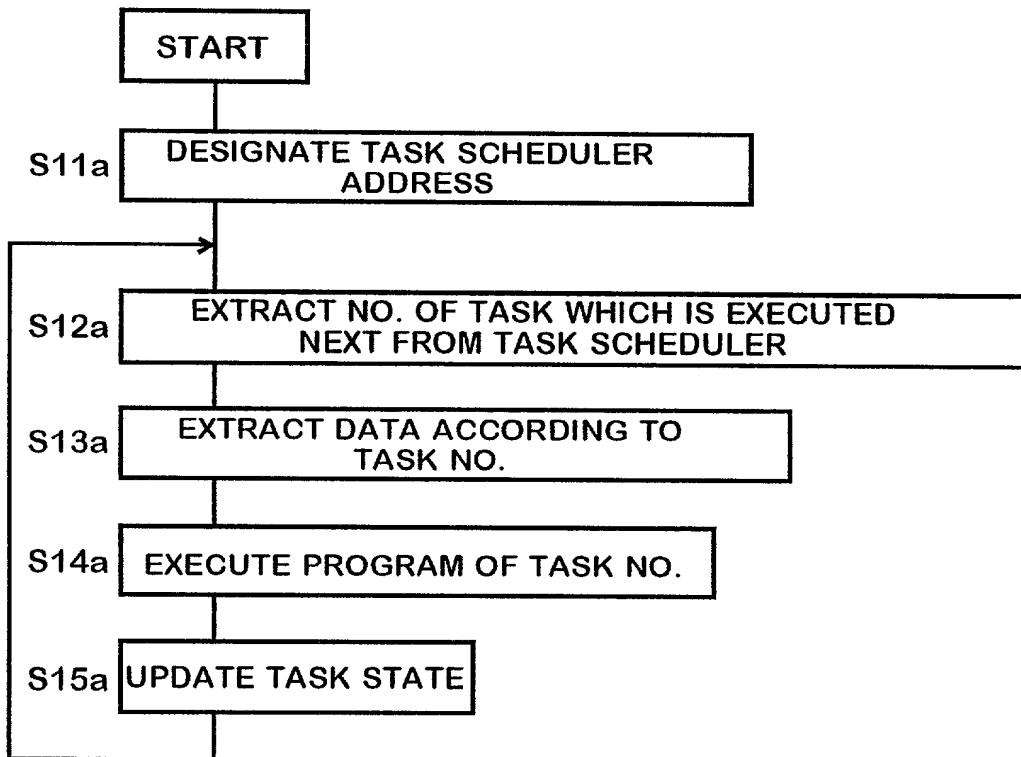


Fig.18B

```
int. *sp;
int. *data[];
sp= TASK_SCHEDULER_ADR; // S11a

while(1) {
    NUM= *sp; // S12a
    state[NUM]=func[NUM](data[NUM]); // S14a&15a&S13a
}
```


FIG. 19A

Fig. 19A

b111111	PROGRAM HEAD ADDRESS
b111100	ADDRESS OF FUNC 1
b111010	ADDRESS OF FUNC 1
b000000	

PROGRAM TABLE

Fig. 19B

b111111	DATA STRUCTURE HEAD ADDRESS
b111100	ADDRESS OF DATA 0
b111010	ADDRESS OF DATA 1
b000000	

DATA TABLE